Table 12 Summary of decision making of the increment for updating CWND by using online training at end-systems of the network

Ref.	RL Technique	Network	Synthetic Dataset	Features	Action-set (*action selection*)	Evaluation Settings	Evaluation Results[a]
TCP-FALA [380]	FALA	WANET	GloMoSim simulation: · Topology: - Random - Dumbbell	States and reward: · IAT of ACKs *(distinguish ACKS and DUPACKs)*	Finite: · 5 actions *(stochastic)*	· 1 input feature · 5 states · 5 actions	To TCP-NewReno[b]: · Packet loss = 66% · Goodput = 29% · Fairness = 20% To TCP-FeW[‡]: · Packet loss = −5% · Goodput = −10% · Fairness = 12%
Learning-TCP [29, 379]	CALA	WANET	Simulation: · ns2 and GloMoSim · Topology: - Chain - Random node - Grid Experimental: · Linux-based · Chain topology	States and reward: · IAT of ACKs	Continuous: · Normal action probability distribution *(stochastic)*	· 1 input feature · 2 states · ∞ actions	To TCP-FeW: · Packet loss = 37% · Goodput = 13% · Fairness = 23% To TCP-FALA: · Packet loss = 28% · Goodput = 36% · Fairness = 14%
TCP-GVegas [219]	Q-learning	WANET	ns-2 simulation: · Topology: - Chain - Random	States: · CWND · RTTz · Throughput Reward: · Throughput	Continuous: · Range based on RTT, throughput, and a span factor *(ϵ-greedy)*	· 3 input features · 3 states · *N/A* actions	To TCP-Vegas: · Throughput = 60% · Delay = 54%
FK-TCPLearning [271]	FKQL	IoT	ns-3 simulation: · Dumbbell topology: - Single source/sink - Double source/sink	States: · IAT of ACKs · IAT of packets sent · RTT · SSThresh Reward: · Throughput · RTT	Finite: · 5 actions *(ϵ-greedy)*	· 5 input features · 10k states · 5 actions · FK approx: - 100 prototypes	To TCP-NewReno: · Throughput = 34% · Delay = 12% To TCPLearning based on pure Q-learning: · Throughput= −1.5% · Delay = −10%
UL-TCP [30]	CALA	Wireless: · Single-hop: - Satellite - Cellular - WLAN · Multi-hop: - WANET	ns-2 simulation: · Single-hop dumbbell · Multi-hop topology: - Chain - Random - Grid	States and reward: · RTT · Throughput · RTO CWND	Continuous: · Normal action probability distribution *(stochastic)*	· 3 input features · 2 states · ∞ actions	For single-hop, to ATL: · Packet loss = 51% · Goodput: = −14% · Fairness = 53% For multi-hop, similar to Learning-TCP
Remy [477]	Own *(offline training)*	· Wired · Cellular	ns-2 simulation: · Wired topology: - Dumbbell - Datacenter · Cellular topology	States: · IAT of ACKs · IAT of packets sent · RTT Reward: · Throughput · Delay	Continuous with 3-dimensions: · CWND multiple · CWND increment · Time between successive sends *(ϵ-greedy)*	· 4 input features · $(16k)^3$ states · 100^3 actions · 16 network configurations	To TCP-Cubic: · Throughput = 21% · Delay = 60% To TCP-Cubic/SFQ-CD: · Throughput = 10% · Delay = 38%
PCC [122]	Own	· Wired · Satellite	Experimental: · GENI · Emulab · PlanetLab	States: · Sending rate Reward: · Throughput · Delay · Loss rate	Finite: · 2 actions of the increment for updating sending rate (not CWND) *(gradient ascent)*	· 3 input features · 4 states · 2 actions	To TCP-Cubic: · Throughput = 21% · Delay = 60%

[a] Average value of improvement ratio. Results vary according to the configured network parameters (e.g. topology, mobility, traffic)
[b] Based on the results from the simulated and experimental evaluations in [29]

memory restrictions of IoT devices, the authors use two function approximation methods: tile coding [435] and Fuzzy Kanerva (FK) [481]. The latter significantly reduces the memory requirements, hence, is incorporated in a modification of TCPLearning, called FK-TCPLearning. Specifically, FK-TCPLearning with a set of 100 prototypes, needs only 1.2% (2.4*KB*) of the memory used by TCPLearning based on pure Q-learning (200*KB*), for storing 50,000 state-action pairs. Furthermore, basic simulations in ns-3 reveal that FK-TCPLearning improves